

L Number	Hits	Search Text	DB	Time stamp
19	33	(link\$3 adj list same linear\$6) and (delet\$3 with (mark\$3 or point\$3))	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 16:57
20	16	(link\$3 adj list same linear\$6) and (delet\$3 near3 (mark\$3 or point\$3))	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 17:02
21	557	(link\$3 adj list) and (delet\$3 near3 (mark\$3 or point\$3))	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 17:02
22	251	(link\$3 adj list) and (delet\$3 near3 (mark\$3))	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 17:12
23	156	(link\$3 adj list) and (delet\$3 near (mark\$3))	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 17:03
24	9	link\$3 adj list same (delet\$3 near (mark\$3))	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 17:12
25	2398	(707/8,100,103.R).CCLS.	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 17:12
26	245	((707/8,100,103.R).CCLS.) and (link\$3 adj list)	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 17:12
27	23	((707/8,100,103.R).CCLS.) and (link\$3 adj list)) and (delet\$3 near3 (mark\$3))	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 17:13

L Number	Hits	Search Text	DB	Time stamp
2	5	(compare adj2 swap adj instruction).ti.	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 16:05
3	8	4584640.URPN.	USPAT	2003/09/25 16:00
4	5	(compar\$3 adj2 swap\$4 adj instruction).ti.	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 16:06
5	24	(compar\$3 adj2 swap\$4).ti.	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 16:08
6	6	(compar\$3 with swap\$4 with data).ti.	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 16:11
8	6	"366" adj "585"	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 16:11
9	13	"366585"	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 16:12
10	2	("5081572").PN.	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 16:31
11	8562	link\$3 adj list	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 17:03
12	275	link\$3 adj list same linear\$6	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 16:32
13	2	link\$3 adj list same linear\$6 same ((non adj block\$3) or (lock adj free))	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 16:39
14	18	link\$3 adj list same linear\$6 and ((non adj block\$3) or (lock adj free))	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 16:52
15	4	link\$3 adj list same linear\$6 and (compar\$3 near2 swap\$4)	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 16:50

16	12	link\$3 adj list same ((non adj block\$3) or (lock adj free)) and (compar\$3 near2 swap\$4)	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB USPAT	2003/09/25 16:52
17	6	(US-4584640-\$ or US-6178473-\$ or US-5081572-\$ or US-6560619-\$ or US-6223335-\$ or US-6581063-\$).did.		2003/09/25 16:55
18	3	((US-4584640-\$ or US-6178473-\$ or US-5081572-\$ or US-6560619-\$ or US-6223335-\$ or US-6581063-\$).did.) and (delet\$3 with (mark\$3 or point\$3))	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 16:57
19	33	(link\$3 adj list same linear\$6) and (delet\$3 with (mark\$3 or point\$3))	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 16:57
20	16	(link\$3 adj list same linear\$6) and (delet\$3 near3 (mark\$3 or point\$3))	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 17:02
21	557	(link\$3 adj list) and (delet\$3 near3 (mark\$3 or point\$3))	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 17:02
22	251	(link\$3 adj list) and (delet\$3 near3 (mark\$3))	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 17:03
23	156	(link\$3 adj list) and (delet\$3 near (mark\$3))	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 17:03
24	9	link\$3 adj list same (delet\$3 near (mark\$3))	USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB USPAT; US-PGPUB; EPO; DERWENT; IBM_TDB	2003/09/25 17:03



Find: lock-free linked list compare and

Documents

Citations

Searching for **lock free linked list compare and swap**.

Restrict to: [Header](#) [Title](#) Order by: [Citations](#) [Hubs](#) [Usage](#) [Date](#) Try: [Amazon](#) [B&N](#) [Google \(RI\)](#) [Google \(Web\)](#) [CSB](#) [DBLP](#)

No documents match Boolean query. Trying non-Boolean relevance query.

1000 documents found. **Only retrieving 250 documents (System busy - maximum reduced)**. Retrieving documents...

**Order: relevance to query.**

[Correction of a Memory Management Method for Lock-Free Data.. - Michael, Scott \(1995\)](#) (Correct) (5 citations)

Correction of a Memory Management Method for **Lock-Free** Data Structures Maged M. Michael Michael L.

Correction of a Memory Management Method for **Lock-Free** Data Structures Maged M. Michael Michael L.

December 1995 Abstract Memory reuse in **link-based lock-free** data structures requires special  
hypatia.dcs.qmw.ac.uk/data/edu/cs.rochester.edu/systems/95.tr599.Memory\_management\_for\_lock-  
free\_data\_structures.ps.gz

[Lock-Free Linked Lists Using Compare-and-Swap - Valois \(1995\)](#) (Correct) (17 citations)

**Lock-Free Linked Lists Using Compare-and-Swap** John D.

**Lock-Free Linked Lists Using Compare-and-Swap** John D.

**Lock-Free Linked Lists Using Compare-and-Swap** John D. Valois

ftp.cs.rpi.edu/pub/valoisj/podc95.ps.gz

[Implementation of Atomic Primitives on Distributed Shared.. - Maged Michael \(1995\)](#) (Correct) (7 citations)

[1] and Tera machines [3] and the primitives for **locking** and unlocking cache lines on the Kendall Square

styles of synchronization (e.g. operations on wait-free and **lockfree** objects, read-write **locks**, priority

primitives fetch and \Phi, compare and **swap**, load **linked**, and store conditional on large-scale

hypatia.dcs.qmw.ac.uk/data/edu/cs.rochester.edu/systems/95.HPCA.Implementation\_of\_general-  
purpose\_atomic\_primitives.ps.gz

[Scalability of Atomic Primitives on Distributed Shared Memory .. - Michael, Scott \(1994\)](#) (Correct) (2 citations)

[1] and Tera machines [3] and the primitives for **locking** and unlocking cache lines on the KSR1 [15]

styles of synchronization (e.g. operations on wait-free and **lock-free** objects, read-write **locks**, priority

of fetch and \Phi, compare and -**swap**, and load **linked**/store conditional. We then analyze the

ftp.cs.rochester.edu/pub/papers/systems/94.tr528.Scalability\_of\_atomic\_primitives.ps.Z

[Real-Time Computing with Lock-Free Shared Objects - Anderson, Ramamurthy, Jeffay \(1997\)](#) (Correct) (6 citations)

Real-Time Computing with **Lock-Free** Shared Objects JAMES H. ANDERSON, SRIKANTH

archi.snu.ac.kr/yhbae/paper/lockfree/Anderson\_TOCS-97.ps.gz

[A Lock-Free Multiprocessor OS Kernel - Massalin, Pu \(1991\)](#) (Correct) (47 citations)

A **Lock-Free** Multiprocessor OS Kernel Henry Massalin and

A **Lock-Free** Multiprocessor OS Kernel Henry Massalin and Calton

www.cs.columbia.edu/~library/TR-repository/reports/reports-1991/cucs-005-91.ps.gz

[Characterizing the Performance of Algorithms for Lock-free Objects - Johnson](#) (Correct)

Characterizing the Performance of Algorithms for **Lock-free** Objects Theodore Johnson Dept. of CIS

ftp.cis.ufl.edu/pub/tech-reports/tr93/tr93-021.ps.gz

[A Framework for Implementing Objects and Scheduling Tasks.. - Anderson, Ramamurthy \(1996\)](#) (Correct) (8 citations)

for Implementing Objects and Scheduling Tasks in **Lock-Free** Real-Time Systems James H. Anderson and

archi.snu.ac.kr/yhbae/paper/lockfree/Anderson\_RTSS-96b.ps.gz

[Practical Considerations for Non-Blocking Concurrent Objects - Version Of](#) (Correct)

or Atomic Exchange. We then evaluate several **locking** strategies that can be used to synthesize a

has shown several practical algorithms for wait-free - a restricted form of non- blocking -

complicated shared data structures such as doubly-linked lists. the time of preemption. If the thread

ftp.cs.cmu.edu/project/mach/doc/published/non-blocking.ps

[Simple, Fast, and Practical Non-Blocking and Blocking.. - Michael, Scott \(1996\)](#) (Correct) (3 citations)

**swap** or load **linked**/store conditional) The two-**lock** concurrent queue outperforms a single **lock** when

each processor. Keywords: concurrent queue, **lock-free**, non-blocking, compare and **swap**,

ftp.cs.rochester.edu/pub/papers/systems/96.PODC.Non-blocking\_and\_blocking\_concurrent\_queue\_algorithms.ps.gz

[Universal Constructions for Multi-Object Operations - Anderson, Moir \(1995\)](#) (Correct) (24 citations)

NC 27599-3175 Abstr ct We present w it-free nd l ck-free univers l c nstructi ns th t ll w per ti ns

[www.cs.unc.edu/~anderson/papers/podc95.ps.Z](http://www.cs.unc.edu/~anderson/papers/podc95.ps.Z)

Transparent Support for Wait-Free Transactions - Mark Moir (1997) (Correct) (7 citations)  
that convert sequential transactions into **lock-free** or wait-free ones. In contrast to some  
Transparent Support for Wait-Free Transactions Mark Moir Department of Computer  
[www.cs.pitt.edu/~moir/Papers/moir-wdag97.ps](http://www.cs.pitt.edu/~moir/Papers/moir-wdag97.ps)

Lock-Free Transactions For Real-Time Systems - Anderson, Ramamurthy, Moir.. (1997) (Correct) (1 citation)  
1 **Lock-Free** Transactions For Real-Time Systems James H.  
[www.cs.unc.edu/~anderson/papers/rtdbbook.ps.Z](http://www.cs.unc.edu/~anderson/papers/rtdbbook.ps.Z)

Intelligent Memory: An Architecture for Lock-Free.. - Seong, Jung, Kim, Yoon.. (Correct)  
1 Intelligent Memory: An Architecture for **Lock-Free** Synchronization Nakun Seong, Naihoon Jung,  
1 Intelligent Memory: An Architecture for **Lock-Free** Synchronization Nakun Seong, Naihoon Jung, Byungho  
or shared objects such as shared stack and the **linked list** [6]and also most of the computations on  
[camars.kaist.ac.kr/publication/paper/mem-ieice.ps.gz](http://camars.kaist.ac.kr/publication/paper/mem-ieice.ps.gz)

Lock-Free Transactions for Real-Time Systems - Anderson, Ramamurthy, Moir.. (1997) (Correct) (1 citation)  
1 **Lock-Free** Transactions For Real-Time Systems Abstract  
[www.cs.pitt.edu/~moir/Papers/rtdbbook.ps](http://www.cs.pitt.edu/~moir/Papers/rtdbbook.ps)

Compile/Run-time Support for Threaded MPI Execution on.. - Tang, Shen, Yang (1999) (Correct)  
-point communication protocol based on a novel **lock-free** queue management scheme. Our experiments on  
-point communication protocol based on a novel **lock-free** queue management scheme. Our experiments on an SGI  
N queues. Each queue is represented by a doubly-linked list. There are three types of operations  
[www.cs.ucsb.edu/TRs/techreports/TRCS98-30.ps](http://www.cs.ucsb.edu/TRs/techreports/TRCS98-30.ps)

Software Transactional Memory - Shavit (1995) (Correct) (31 citations)  
translating sequential object implementations to **lock-free** ones based on implementing a k-word  
sequential object implementations to **lock-free** ones based on implementing a k-word compare&swap  
inflexible and is at best on the level of a Load **Linked**/Store Conditional operation on a single word.  
[wilma.cs.brown.edu/courses/cs295h/stm.ps](http://wilma.cs.brown.edu/courses/cs295h/stm.ps)

*First 20 documents* [Next 20](#)

Try your query at: [Amazon](#) [Barnes & Noble](#) [Google \(RI\)](#) [Google \(Web\)](#) [CSB](#) [DBLP](#)

CiteSeer - [citeseer.org](http://citeseer.org) - [Terms of Service](#) - [Privacy Policy](#) - Copyright © 1997-2002 [NEC Research Institute](#)



US Patent &amp; Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☐ The Guide ☒ The ACM Digital Library

+linked +list compare

action

THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used linked list compare

Found 24,108 of 121,259

Sort results by

relevance

Display results

condensed form

☒ Save results to a Binder☒ Search Tips☐ Open results in a new windowTry an [Advanced Search](#)Try this search in [The ACM Guide](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐1 [Lock-free linked lists using compare-and-swap](#)

John D. Valois

August 1995 **Proceedings of the fourteenth annual ACM symposium on Principles of distributed computing**Full text available: [pdf\(902.29 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)2 [A comparative study of parallel and sequential priority queue algorithms](#)

Robert Rönngren, Rassul Ayani

April 1997 **ACM Transactions on Modeling and Computer Simulation (TOMACS)**, Volume 7 Issue 2Full text available: [pdf\(640.10 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)3 [C++ dynamic arrays vs. linked lists](#)

Marsha Zaidman

September 1996 **ACM SIGCSE Bulletin**, Volume 28 Issue 3Full text available: [pdf\(267.30 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)4 [Split-ordered lists: lock-free extensible hash tables](#)

Ori Shalev, Nir Shavit

July 2003 **Proceedings of the twenty-second annual symposium on Principles of distributed computing**Full text available: [pdf\(1.06 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)5 [Xanalogical structure, needed now more than ever: parallel documents, deep links to content, deep versioning, and deep re-use](#)

Theodor Holm Nelson

December 1999 **ACM Computing Surveys (CSUR)**Full text available: [pdf\(787.72 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)6 [Simple, fast, and practical non-blocking and blocking concurrent queue algorithms](#)

Maged M. Michael, Michael L. Scott

May 1996 **Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing**Full text available: [pdf\(860.15 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)7 [A simple and correct shared-queue algorithm using Compare-and-Swap](#)

Janice M. Stone

November 1990 **Proceedings of the 1990 ACM/IEEE conference on Supercomputing**Full text available: [pdf\(929.76 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)8 [istributed c mputing: N nbl cking k-c mp re-single-sw p](#)

Victor Luchangco, Mark Moir, Nir Shavit

June 2003 **Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures**

Full text available:  pdf(197.17 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

9 Computing curricula 2001

September 2001 **Journal on Educational Resources in Computing (JERIC)**

Full text available:  pdf(613.63 KB)  
 html(2.78 KB) Additional Information: [full citation](#), [references](#), [index terms](#)

10 Implementing wait-free objects on priority-based systems

James H. Anderson, Srikanth Ramamurthy, Rohit Jain


August 1997 **Proceedings of the sixteenth annual ACM symposium on Principles of distributed computing**

Full text available:  pdf(1.19 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

11 Session 3: High performance dynamic lock-free hash tables and list-based sets

Maged M. Michael

August 2002 **Proceedings of the fourteenth annual ACM symposium on Parallel algorithms and architectures**

Full text available:  pdf(238.11 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

12 Shadowed management of free disk pages with a linked list

Matthew S. Hecht, John D. Gabbe

December 1983 **ACM Transactions on Database Systems (TODS)**, Volume 8 Issue 4

Full text available:  pdf(877.39 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

13 Doubly Linked Lists and the Abstract Data Type

Carl Nobile


May 1998 **Linux Journal**

Full text available:  html(19.37 KB) Additional Information: [full citation](#), [abstract](#), [index terms](#)

14 A Two-list method for synchronization of event driven simulation

John H. Blackstone, Gary L. Hogg, Don T. Phillips

March 1981 **Proceedings of the 14th annual simulation symposium**

Full text available:  pdf(387.11 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

15 The software lookaside buffer reduces search overhead with linked lists

Gerald Bozman


March 1984 **Communications of the ACM**, Volume 27 Issue 3

Full text available:  pdf(360.12 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

16 Maintaining order in a linked list

Paul F. Dietz


May 1982 **Proceedings of the fourteenth annual ACM symposium on Theory of computing**

Full text available:  pdf(308.26 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

17 Efficient implementation of event sets in Time Warp

Robert Rönngren, Rassul Ayani, Richard M. Fujimoto, Samir R. Das


July 1993 **ACM SIGSIM Simulation Digest , Proceedings of the seventh workshop on Parallel and distributed simulation**, Volume 23 Issue 1

Full text available:  pdf(900.65 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

18 Animation of Java linked lists

Herbert L. Dershem, Ryan L. McFall, Ngozi Uti

February 2002 **ACM SIGCSE Bulletin , Proceedings of the 33rd SIGCSE technical symposium on Computer science education**, Volume 34 Issue 1

Full text available:  pdf(634.92 KB) Additional Information: [full citation](#), [abstract](#), [references](#)

**19** [A Survey of Indexing Techniques for Sparse Matrices](#)

Udo W. Pooch, Al Nieder

June 1973 **ACM Computing Surveys (CSUR)**, Volume 5 Issue 2

Full text available:  pdf(1.86 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

**20** [Referencing lists by an edge](#)

David S. Wise

June 1976 **Communications of the ACM**, Volume 19 Issue 6

Full text available:  pdf(458.78 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2003 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)